A tractable tree distribution parameterized by clade probabilities and its application to Bayesian phylogenetic point estimation

Lars Berling[1,2], Jonathan Klawitter[3], Remco Bouckaert[3], Dong Xie[3], Alex Gavryushkin[1,2], and Alexei J. Drummond[3,⋆]

[1] School of Mathematics and Statistics, University of Canterbury, New Zealand
[2] Biomathematics Research Centre, University of Canterbury, New Zealand
[3] University of Auckland, New Zealand
⋆ a.drummond@auckland.ac.nz

Abstract. Bayesian phylogenetic analysis with MCMC algorithms generates an estimate of the posterior distribution of phylogenetic trees in the form of a sample of phylogenetic trees and related parameters. The high dimensionality and non-Euclidean nature of tree space complicates summarizing the central tendency and variance of the posterior distribution in tree space. Here we introduce a new tractable tree distribution and associated point estimator that can be constructed from a posterior sample of trees. Through simulation studies we show that this point estimator performs at least as well and often better than standard methods of producing Bayesian posterior summary trees. We also show that the method of summary that performs best depends on the sample size and dimensionality of the problem in non-trivial ways.

1 Introduction

One of the main inference paradigms in phylogenetics is Bayesian inference using Markov Chain Monte Carlo (MCMC) [6, 17, 23]. The central parameter of phylogenetic models is the tree topology describing the evolutionary relationships for a set of taxa. Bayesian inference is based on a statistical model that describes the probability of a set of sequences given a phylogenetic tree, consisting of a topology with associated branch lengths and model parameters. The MCMC algorithm iteratively samples a state space that, if set up with appropriate length and sampling interval, returns a sample that is a representation of the true underlying posterior distribution. In the case of phylogenetic MCMC algorithms, the output of such an analysis is a sample of phylogenetic trees, typically numbering in the thousands.

In a phylogenetic analysis, the posterior distributions of many continuous parameters (e.g. kappa, base frequencies, molecular clock rate, population size) are easily summarised by considering statistics of the marginal distribution of the parameter of interest from the samples obtained by MCMC. On the other hand, one of the most crucial parameters – the tree topology – is a discrete parameter whose central tendency and variance are harder to characterise due to the high-dimensional and non-Euclidean nature of tree space [4,5,11]. It has thus become standard practise to employ summary or consensus tree methods to condense the output into a single tree [14]. This single tree, which should be regarded as a Bayesian *point estimate*, is then used for further representation and interpretation of an analysis. Despite considerable efforts dedicated to the development of point estimators [9], it remains unclear which method performs best for summarising tree posteriors. Most point estimators construct a tree in two steps [14]: First, a tree topology is constructed or selected, and, second, this discrete topology is then annotated with branch lengths. In this paper we focus on the first step, the construction of a rooted binary tree topology.

The predominant challenge for many point estimators is the complexity of the tree space they are operating on. This is particularly the case for methods trying to compute a mean in a high-dimensional, non-Euclidean space such as the BHV space [2,5,8] or a space induced by rearrangement operations [4]. While good progress has been made, these methods can suffer from stickiness and are not tractable yet for large problems [4,8]. The two most popular methods in practice thus operate only on the sampled trees. First, consensus methods focus on finding a consensus among the given trees. The prevalent variant is the *greedy majority-rule consensus (greedy consensus) tree*, which builds up a tree by including clade after clade greedily (i.e., more frequent clades first) that are compatible with the current tree; ties are broken arbitrarily [9]. Consensus methods are however prone to polytomies (i.e., parts of the tree remain unresolved) and finding the most resolved greedy MRC tree is an NP-hard problem [24]. Second, the *maximum clade credibility (MCC) tree* picks the tree from the sample distribution with maximum product of (Monte Carlo) clade probabilities. While the computation of the MCC tree is fast and efficient, it comes at a cost in accuracy due to the restriction to the sampled trees.

A good estimate of the tree distribution is still needed for questions concerning, for example, the credibility set of trees and the information content (entropy) [19] as well as for applications such as Bayesian concordance analysis (BCA) [1]. Introduced by Höhna and Drummond [16] and improved by Larget [18], the *conditional clade distribution (CCD)* offers an advanced estimate of the posterior probability distribution of tree space. Based on simple statistics of the sample, it provides normalized probabilities of all represented trees and allows direct sampling from the distribution. CCDs have for example been used to measure the information content and detect conflict among data partitions [19], for species tree–gene tree reconciliation [25], and for guiding tree proposals for MCMC runs [16]. Constructing the CCD and performing these tasks can be done efficiently [18,19].

In this paper we extend the applicability of CCDs by introducing a new parametrization for CCDs and describing fixed-parameter tractable algorithms to compute the tree with highest probability. We demonstrate the usefulness of the new distribution and these new point estimates for Bayesian phylogenetics by comparing them to existing methods in simulation studies.

2 Methods

In this section, we first discuss tractable tree distributions and define CCDs with two different parametrizations. We then recall the definitions of the MCC and greedy consensus tree and show how CCDs give rise to new point estimators. Lastly, we describe the datasets we generated for our experiments. Throughout, we write tree instead of tree topology and further assume that all our trees are rooted and, unless mentioned otherwise, are binary.

2.1 Tractable Tree Distributions

We consider a probability distribution over a set of trees (on the same taxa) a *tractable tree distribution* if some common tasks can be performed efficiently in practice. Example tasks are computing the probability of a tree and retrieving the tree with maximum probability. As the main quality criteria for a tractable tree distribution we consider its *accuracy*, that is, how well it estimates the probability of trees, in particular of those in the 95% credibility set. In simulation studies we can also test whether a distribution contains the true tree. Another desideratum is a high *representativeness* as a distribution should represent the trees with non-negligible posterior probability but not more. If we generate a type of distribution for the same data multiple times, we can consider the *precision* and the *stability*, that is, how much the probabilities of trees and how much the accuracy change, respectively. Since CCDs, as we see below, are deterministically generated from samples, we can only measure these indirectly through samples from different MCMC runs.

A simple example distribution is the set of sampled trees from an MCMC run; we call this a *sample distribution*. It offers Monte Carlo probabilities and while some tasks can be performed efficiently, it has quite low accuracy, poor representativeness, and is in general not stable. In fact, since the space of trees increases super-exponentially with the number of taxa, a sample on several thousand trees typically misses the majority of trees with non-negligible posterior probability even for moderate size problems.

Reintroducing the concept of a CCD, we first define a graph, which we call a *forest network*, capable of representing a larger number of trees. Assigning probabilities to certain vertices (or edges), we obtain a *CCD graph*. The version of a CCD by Larget [18] is one possible parametrization of a CCD based on clade split frequencies; we call this a CCD1. Our new parametrization, CCD0, is based on clade frequencies. We also show how to efficiently sample trees from a CCD and how dynamic programming allows efficient computation of values such as the number of trees and its entropy.

Forest network. Let X be a set of n taxa. A *forest network* N on X is a rooted bipartite digraph with vertex set $(\mathcal{C}, \mathcal{S})$ that satisfies the following properties:

- Each $C \in \mathcal{C}$ represents a *clade* on X. So for each $C \in \mathcal{C}$, we have $C \subseteq X$; for each taxon $\ell \in X$, $\{\ell\} \in \mathcal{C}$, and also $X \in \mathcal{C}$.

- Each $S \in \mathcal{S}$ represents a *clade split*. So each $S \in \mathcal{S}$ has degree three with one incoming edge (C, S) and two outgoing edges (S, C_1), (S, C_2) such that $C_1 \cup C_2 = C$, $C_1 \cap C_2 = \emptyset$ for some $C_1, C_2, C \in \mathcal{C}$. Then $S = \{C_1, C_2\}$ and S is a clade split of C.
- Each non-leaf clade has outdegree at least one and each clade except X has indegree at least one.

Note that X is the root of N, the taxa in X are the leaves of N, and each non-leaf clade has at least one clade split. We use terms such as *child* and *parent* naturally to refer to relations between vertices of N, e.g., each clade split S has a parent clade C. When talking about multiple graphs, we let $\mathcal{C}(N)$ and $\mathcal{S}(N)$ denote the clades and clade splits, respectively, of N. For a (rooted binary phylogenetic) tree T on X, we use analogous definitions for $\mathcal{C}(T)$ and $\mathcal{S}(T)$ (each pair of sibling clades in T forms a clade split of T). For a clade C, we define $\mathcal{S}(C)$ as the set of child clade splits of C.

A tree T is *displayed* by N if each clade split of T is in $\mathcal{S}(N)$, i.e., $\mathcal{S}(T) \subseteq \mathcal{S}(N)$. For a clade C define $N(C)$ as the restriction of N to C, that is, the forest subnetwork rooted at C containing all vertices reachable from C. Analogously, for $S \in \mathcal{C}(N)$, we can define the forest subnetwork $N(S)$ of N that is rooted at the parent clade C of S but contains only S as child of C and all vertices reachable from S. Note that, for a clade split $\{C_1, C_2\}$ of X, network N contains all trees composed (amalgamated) of one subtree from $N(C_1)$ and one subtree from $N(C_2)$; this holds recursively. Hence, a forest network is suitable to represent huge numbers of trees when all combinations of subtrees are included.

CCD graph. In order to turn a forest network into a tree distribution, we need to be able to compute a probability for a tree T. Larget [18] suggested to use the product of clade split probabilities over all clade splits in $\mathcal{S}(T)$ as the probability of T. We define a *CCD graph* as a forest network G where each clade split S in $\mathcal{S}(G)$ has an assigned probability $\Pr(S)$ such that, for each clade $C \in \mathcal{C}(G)$, we have $\sum_{S \in \mathcal{S}(C)} \Pr(S) = 1$. In other words, we can randomly pick a clade split at C. From Larget [18, Appendix 2] we then get that G represents a tree distribution. So for a tree T displayed by G, we have

$$\Pr(T) = \prod_{S \in \mathcal{S}(T)} \Pr(S)$$

and, for any other tree T', we have $\Pr(T') = 0$. Furthermore, the sum of probabilities of all trees displayed by G is one. We now show how CCD1 and CCD0 assign probabilities based on observed clade split and clade frequencies, respectively.

CCD1, observed clade splits. CCD1 is a tree distribution over the space of trees on a fixed set of taxa X based on a CCD graph with clade split probability obtained as follows. Let $\mathcal{T} = \{T_1, \ldots, T_k\}$, a (multi-)set of trees on X, e.g., the samples of an MCMC run. Let $\mathcal{C}$ and $\mathcal{S}$ be the sets of clades and clade splits appearing in $\mathcal{T}$, respectively. Then let G be the forest network induced by $\mathcal{T}$, that is, G has vertex set $(\mathcal{C}, \mathcal{S})$ and edges naturally induced by the clade splits $\mathcal{S}$ (we know the two child clades and the parent clade of each clade split). Furthermore, we assign clade split probabilities as follows to turn G into a CCD graph. For a clade $C \in \mathcal{C}$ and a clade split $S \in \mathcal{S}$, let $f(C)$ and $f(S)$ denote the frequencies of C and S appearing in the sample $\mathcal{T}$, respectively. Note that

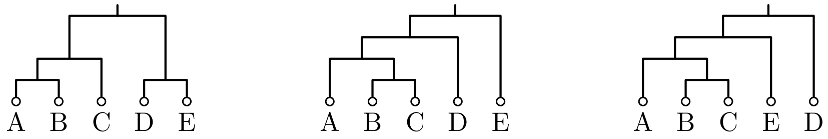
1. $f(S) \leq f(C)$ for all pairs of S, C with $S \in \mathcal{S}(C)$;
2. $\sum_{S \in \mathcal{S}(C)} f(S) = f(C)$ for a non-leaf clade C;
3. $f(X) = k$ and, for each $\ell \in X$, $f(\{\ell\}) = k$.

The *conditional clade probability (CCP)* $\Pr(S)$ of a clade split S is defined as the ratio of S being the split of C in the posterior sample, i.e.,

$$\Pr(S) = f(S)/f(C)$$

Note that $\sum_{S \in \mathcal{S}(C)} \Pr(S) = 1$ and $\Pr(S) = 1$ if $S \in \mathcal{S}(\{a, b\})$ for some leaves a, b. The resulting CCD graph is what we call a *CCD1*, the conditional clade distribution induced by the probability distributions of clade splits.

Example. Let us consider the example shown in Fig. 1 where the posterior samples consists of three trees with the first being sampled three times, and the others twice each. Observe that the root clade ABCDE is split in three different ways, namely, ABC|DE, ABCD|E, and ABCE|D. The probabilities of these three clades splits are Pr(ABC|DE) = 3/7, Pr(ABCD|E) = 2/7, and Pr(ABCE|D) = 2/7. Furthermore, the clade ABC is split in two different ways with probabilities Pr(AB|C) = 3/7 and Pr(A|BC) = 4/7. All other clades are trivial or are only split in one way, e.g., the clade ABCD is always split into ABC|D, so Pr(ABC|D) = 1/1.



(a) Tree 1, sampled three times.

(b) Tree 2, sampled twice.

(c) Tree 3, sampled twice.

Fig. 1: Example of a posterior sample of size seven consisting of three different trees. Only the clades ABCDE and ABC are split in multiple ways.

The resulting CCD contains 6 different trees – the three sampled trees as well as three unsampled trees, two of which are shown in Figs. 3b and 3c. Note that the tree sampled most often still has the highest probability, with $3/7 \cdot 3/7 = 9/49$, among the sampled trees, as the other two trees have a probability of

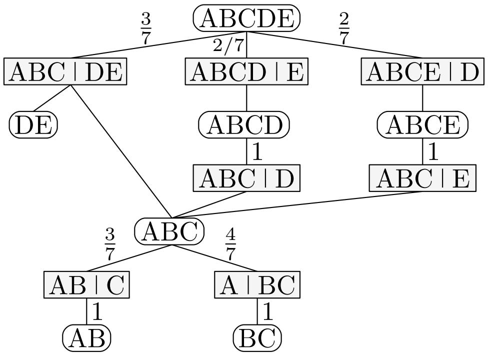
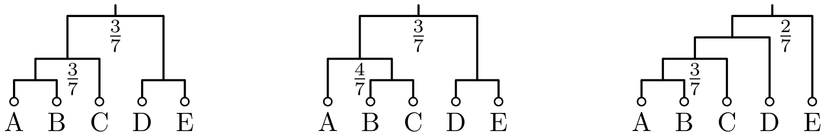


Fig. 2: The CCD graph of the CCD1 corresponding to the sampled trees in Fig. 1.

$1/7 \cdot 4/7 = 4/49$ each. Furthermore, the unsampled tree containing the most frequent clade split ABC|DE of the root clade and the most frequent clade split A|BC of ABC, has a higher probability of $3/7 \cdot 4/7 = 12/49$.



(a) The sampled Tree 1 has probability $3/7{\cdot}3/7 = 9/49$.

(b) The unsampled tree has the maximal probability of $3/7 \cdot 4/7 = 12/49$.

(c) This unsampled tree has probability $2/7{\cdot}3/7 = 6/49$.

Fig. 3: The CCD of the example posterior sample of Fig. 1 contains unsampled topologies and smoothens the probabilities.

Remark. When computing the CCD1 graph, it is important that the tree sample $\mathcal{T}$ does not contain outlier trees that should have been discarded as burnin. Suppose otherwise, that there is an outlier tree T that does not share any clades (except X and the taxa) with the other trees in $\mathcal{T}$. Then X has one clade split S_T corresponding to T with $\Pr(S_t) = \frac{1}{k}$; all other non-leaf clades of T have only one clade split and so probability 1. Therefore, $\Pr(T) = \frac{1}{k}$ and is thus vastly overestimated. However, it is possible to built a simple heuristic to detect such outliers: Does removing a tree T from the CCD1 and thus decreasing clade and clade split frequencies by one, significantly change the probability of T. Nonetheless, this behaviour should be kept in mind when working with CCD1 and in particular when $\mathcal{T}$ contains only few different trees.

CCD0, observed clades. For the new CCD, our goal is to have a distribution where the probability of a tree is based on the product of its clades' frequencies.

We could derive the probability of a clade C from a posterior sample on k trees as $\Pr'(C) = f(C)/k$. This does in general however not yield a distribution as the tree probabilities do not sum to one. Since for complex problems even large samples may not contain all plausible clade splits, we have as another feature for CCD0 that we also include (some) non-observed clade splits.

Using a forest network, we show how to build a CCD based on a function $F\colon 2^X \to [0,1]$ with $F(\{\ell\}) = F(X) = 1$. For example, for CCD0 we use $F(C) = \Pr'(C)$; so we show something more general than needed for CCD0. For a tree T on X, define $F(T) = \prod_{C \in \mathcal{C}(T)} F(C)$, i.e., the product for the clades of T. Let $\mathcal{C}$ be the subset of 2^X containing all clades with non-zero scores ($F(C) > 0$). Let $\mathcal{S}$ be the set of all possible clade splits that can be formed from $\mathcal{C}$, that is, for any three clades $C_1, C_2, C \in \mathcal{C}$ with $C_1 \cup C_2 = C$ and $C_1 \cap C_2 = \emptyset$, we have $\{C_1, C_2\} \in \mathcal{S}$. Let G be the forest network based on $\mathcal{C}$ and $\mathcal{S}$. (In the example above, there are no additional clade splits besides the observed ones for CCD1.)

Let $F_+\colon \mathcal{C} \to \mathbb{R}$ be the function defined as $F_+(C) = \sum_{T \in G(C)} F(T)$, i.e., the sum of F over all trees rooted at C. This can be computed recursively with the formula

$$F_+(C) = F(C) \cdot \sum_{\{C_1,C_2\} \in \mathcal{S}(C)} F_+(C_1) F_+(C_2)$$

with base case $F_+(\{\ell\}) = 1$ for each $\ell \in X$. Note that $F_+(X)$ is the sum over all trees on X and hence $\alpha = 1/F_+(X)$ is the global normalization factor turning the score $F(T)$ of a tree T into a probability $\Pr(T) = \alpha F(T)$. Populating the clades in G with F, we can compute $F_+(X)$ efficiently (more precisely, fixed-parameter tractable in the number of clades and clade splits with non-zero scores). Moreover, we show next that we can also define probability distributions for the clade splits and hence turn G into a CCD graph.

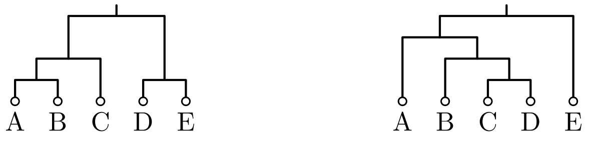
For a clade C, with clade splits $\mathcal{S}(C)$, define the normalization factor α_C for C as $\alpha_C = 1/\sum_{\{C_1,C_2\} \in \mathcal{S}(C)} F_+(C_1)F_+(C_2)$. Then the probability of a clade split $\{C_1, C_2\}$ of C is $\Pr(\{C_1, C_2\}) = \alpha_C F_+(C_1) F_+(C_2)$. We have to show that this gives the same probability for a tree T:

$$\begin{aligned}
\Pr(T) &= \prod_{\{C_1,C_2\} \in \mathcal{S}(T)} \Pr(\{C_1, C_2\}) \\
&= \prod_{\substack{\{C_1,C_2\} \in \mathcal{S}(T) \\ C_1 \cup C_2 = C}} \alpha_C F_+(C_1) F_+(C_2) \\
&\overset{(1)}{=} \prod_{\substack{\{C_1,C_2\} \in \mathcal{S}(T) \\ C_1 \cup C_2 = C}} \frac{\alpha_C F(C_1) F(C_2)}{\alpha_{C_1} \alpha_{C_2}} \\
&\overset{(2)}{=} \alpha \prod_{C \in \mathcal{C}(T)} F(C) = \alpha F(T)
\end{aligned}$$

where for (1) we use that $F_+(C) = F(C)/\alpha_C$, and for (2) observe that $\alpha = \alpha_X$ and that all other α_C cancel out and are 1 for singleton clades. Hence, using $F(C) = \mathrm{Pr}'(C)$, we get that G is a CCD graph as desired, our new tree distribution CCD0.

Both CCD0 and CCD1 are estimates of the true posterior tree distribution. Their models assume that clades/clade splits in one part of a tree behave independent of other clades. So a CCD smoothens the probabilities of a sample distribution by moving probability of overrepresented sampled trees to trees that have not been sampled, but whose clades/clade splits appear within the samples.

Example, continued. Note that the three sampled trees from Fig. 1 result in the same CCD graph (Fig. 2) for CCD0 and CCD1 as no potential pair of a child clades can be combined into an unobserved parent clade. In contrast, in the example in Fig. 4, CCD0 and CCD1 are different as CCD0 contains the clade split AB|CD (we observe clades AB, CD, and ABCD) but CCD1 does not (we do not observe this clade split).



(a) Topology 1 (b) Topology 2

Fig. 4: For this sample of trees, the CCD graph of CCD0 and CCD1 differ.

Utilizing CCDs. With the CCD graph as data structure underlying CCD0 and CCD1, we can efficiently sample and compute interesting values over a whole CCD. To sample a tree from a CCD, starting at the root clade X, pick a clade split $\{C_1, C_2\}$ among $\mathcal{S}(X)$ based on their probabilities; then proceed in the same fashion with C_1 and C_2 until a fully resolved tree is obtained.

We can also use dynamic programming to compute values such as the number of different trees (topologies) and the entropy of a CCD, or (as explained below) find the tree with maximum probability. For example, to compute the number of different trees in a CCD graph G, for a clade C, let $\mathrm{t}(C)$ be the number of different trees in $G(C)$. For a leaf ℓ, we have $\mathrm{t}(\ell) = 1$, and for any other clade, we can use the following recursive formula:

$$\mathrm{t}(C) = \sum_{\{C_1, C_2\} \in \mathcal{S}(C)} \mathrm{t}(C_1)\,\mathrm{t}(C_2)$$

Using dynamic programming, we compute these values bottom-up through G. Then $\mathrm{t}(X)$ is the total number of different trees in G. Note that this calculation takes linear time in the number of clades and clade splits.

Analogously, we can compute the entropy of the CCD by computing, for each clade C, the entropy of $G(C)$; let $H^\star(C)$ denote this value. We can then use the formula by Lewis et al. [19]:

$$H^\star(C) = \sum_{\substack{S \in \mathcal{S}(C) \\ S = \{C_1, C_2\}}} - \Pr(S)\big(\log \Pr(S) - H^\star(C_1) - H^\star(C_2)\big)$$

The entropy of the CCD is then $H = H^\star(X)$. Note that $\exp(-H)$ is the average probability of a tree in the CCD and we can define $N_e = \exp(H)$ as the *number equivalent* – the effective number of distinct topologies in the distribution.

2.2 Point Estimators

We recall the definitions of the two most commonly used point estimators and define new point estimators based on CCD0 and CCD1. Let $\mathcal{T}$ be again a tree sample on k trees for which we can compute the frequencies for trees, clades, and clade splits.

MCC tree. Let $\Pr_{\mathrm{CC}}(C)$ denote the clade credibility (Monte Carlo probability) of clade C, i.e., $\Pr_{\mathrm{CC}}(C) = f(C)/k$. The *clade credibility* $\Pr_{\mathrm{CC}}(T)$ of a tree $T \in \mathcal{T}$ is the product of its clades' clade credibilities:

$$\Pr_{\mathrm{CC}}(T) = \prod_{C \in \mathcal{C}(T)} \Pr_{\mathrm{CC}}(C)$$

The *maximum clade credibility (MCC) tree* is the tree T in $\mathcal{T}$ that maximizes $\Pr_{\mathrm{CC}}(T)$. Note that the MCC tree is restricted to be from the sample.

Greedy majority-rule consensus tree. Let $C_1, \ldots, C_m$ be the nontrivial clades appearing in $\mathcal{T}$ ordered by decreasing frequency; ties are broken arbitrarily. Starting with a star tree T' with root X and leaves $\{\ell\}$, $\ell \in X$, we process the clades in order. For the next clade C_i, we test whether C_i is compatible with current tree T', that is, whether there is a clade (vertex) C containing C_i in T' and with no child clade C' of C containing or properly intersecting C_i. If we find such a clade C, we refine T' by making C_i a new child of C and making all child clades of C that are contained in C_i child clades of C_i. After C_m, the resulting tree is the *greedy majority rule consensus (greedy consensus) tree*. For n taxa and k trees, the greedy consensus tree can be computed in $\mathcal{O}(k^2 n)$ time or $\mathcal{O}(kn^{1.5} \log n)$ time [12, 24], or in $\tilde{\mathcal{O}}(nk)$ time [26] ($\tilde{\mathcal{O}}(\cdot)$ ignores logarithmic factors).

CCD-based point estimators. For a CCD (CCD0 or CCD1), we call the tree T with maximum probability $\Pr(T)$ in the CCD the *CCD-MAP tree*. Using the dynamic program for CCDs explained above, we can find the CCD-MAP tree efficiently as follows. Let $\Pr^\star(C)$ denote the maximum probability of any subtree rooted at clade C. With $\Pr^\star(\ell) = 1$ for a leaf ℓ, we can compute $\Pr^\star(C)$ with the following formula:

$$\Pr^\star(C) = \max_{\{C_1, C_2\} \in \mathcal{S}(C)} \left\{ \quad \Pr(C_1, C_2 \mid C) \cdot \Pr^\star(C_1) \cdot \Pr^\star(C_2) \quad \right\} \tag{1}$$

The maximum probability of any tree in the CCD is then given by $\Pr^\star(X)$. The tree T achieving this can be obtained in the same fashion (or classic dynamic programming backtracking).

Note that the CCD0-MAP tree is based on the same criteria as the MCC tree but the choice is not restricted to the sample. Further note that the greedy consensus greedily picks clades based on their clade credibility. We combine these two ideas into another point estimator for CCD0. The *CCD0-MSCC tree* ('S' for 'sum') is the tree in the CCD0 that maximizes the sum of clade credibilities.

When annotating a tree T obtained with a CCD with clade support, an alternative to the Monte Carlo probabilities from the MCMC run is to use the probability of each clade of T to appear in a tree of the CCD. These values can be computed efficiently with the CCD graph. We do not expect these values to vary significantly in practice.

2.3 Datasets

We performed well-calibrated simulation studies [20] using the LinguaPhylo packages LPhyStudio and LPhyBEAST [10] and BEAST2 [6] to obtain posterior samples. We used both Yule tree and time-stamped coalescent simulations. (See Appendix A for graphical models.)

For our Yule tree simulations we generated two sets of 250 trees and alignments with 10 and 20 (n) taxa, as well as 100 trees and alignments with 50, 100, 200 and 400 taxa. For all simulations (except $n = 20$) the birth rate of the Yule [27] process was fixed to 25.0 (12.5 for $n = 20$). For the substitution model, we used the HKY+G model [13]. The shape parameter for the gamma distribution of site rates was modelled using a log-normal distribution, with a mean in log space of -1.0 and a standard deviation in log space of 0.5. The transition/transversion rate ratio (κ) also followed a log-normal distribution, with a mean in log space of 1.0 and a standard deviation in log space of 1.25. The nucleotide base frequencies were independently simulated for each replicate from a Dirichlet distribution with a concentration parameter array of [5.0, 5.0, 5.0, 5.0]. The length of the sequence alignments was 300 sites (600 sites for $n = 20$) and the mutation rate was fixed at 1.0, so that divergence ages were in units of substitutions per site.

In our time-stamped coalescent [22] simulations, we generated 100 phylogenetic trees and alignments for each of four different taxa sizes n: 40, 80, 160, and 320. Each tree coalescent process had a population size parameter (θ) drawn from a log-normal distribution with a mean in log space of -2.4276, representing a mean in real space of approximately 0.09, and a standard deviation in log space of 0.5. The alignments consisted of 250 sites each. The youngest leaf was assigned age 0. The remaining leaf ages were distributed uniformly at random between 0 and 0.2. All other parameters were as in the Yule simulations.

We refer to the resulting datasets with `Coal40`, ..., `Coal320`, `Yule10`, ..., `Yule400`. For each simulation, we ran 2 chains with BEAST2 to obtain tree samples with 35k trees (50k trees for $n = 10$ and 20). In all cases, the chains were checked to have run sufficiently long to ensure convergence, and excess burnin was discarded.

3 Results

We have presented a new tree distribution, CCD0, and introduced new point estimators. We now apply both CCD0 and CCD1 to the datasets described above to evaluate their point estimators and their performance as tractable tree distributions.

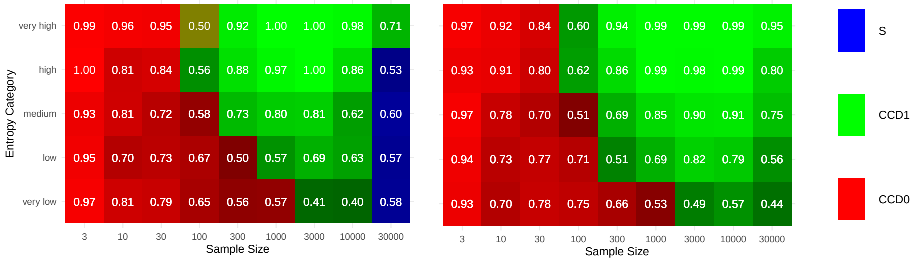
3.1 Tree Distributions

To evaluate the accuracy and precision of CCD0, CCD1, and sample distributions, we used the datasets `Yule10` and `Yule20`. For each simulation, we combined the 50k trees from the two runs into one sample distribution of 100k trees, which acts as our (reference) *golden distribution*. These inference problems are relatively easy, and therefore, the probability of each tree (in particular, the high probability trees) is quite accurately estimated by the golden distributions.

For each simulation and each of the two runs, we used subsamples of size 3, 10, 30, 100, 300, 1k, 3k, 10k, and 30k (which we call *subsimulations*) to generate a CCD0, a CCD1, and a sample distribution each. For each tree T in the golden distribution, we then calculated the probability of T in each of the six distributions. Comparing these to the golden probabilities, we use different statistical measures to evaluate the accuracy of each distribution.

Accuracy. For each subsimulation, we computed the mean absolute error (MAE) of tree probabilities (mean over all trees of the golden distribution) for each distribution. Note that the MAE weights the accuracy on high-probability trees more compared to lower probability trees. We then counted how often each distribution type had the lowest MAE, their number of *wins*. We further divided the simulations into five equal-sized groups (each of size 100) based on their entropy [19], that is, the sum of $-\Pr(T)\log\Pr(T)$ over all trees in the golden distribution. (For Yule10 the entropy bounds are set by 0.41 – 1.76 – 2.5 – 3.25 – 4.30 – 7.68 with means of 1.20, 2.09, 2.84, 3.67, 5.30 and for Yule20 they are 0.09 – 2.29 – 3.22 – 4.03 – 5.08 – 7.73 with means of 1.70, 2.82, 3.61, 4.52, 5.93.) Heatmaps of the wins in these categories for `Yule10` and `Yule20` are shown in Fig. 5, where each tile is colored by the distribution that has the majority of wins and its win-% is given.

We observe that there are three regimes based on the sample size: Roughly, from 3 to about 100 samples, CCD0 is the most accurate method; from 100 to 10k samples, CCD1 gives the best estimates; for the largest samples, the sample distributions catch up with CCD1. A lower entropy seems to prolong the dominance of CCD0. The boundaries of the regimes also vary with the problem size. The experiment confirms the regimes we expected: CCD0 is the simplest model and quickly provides a good estimate; CCD1 has more parameters, so needs longer to be saturated, whereas CCD0 then starts to show its bias. In the long run, the sample distributions provide the best estimate, which we can still observe for 30k trees for `Yule10`, but no longer for `Yule20`.



(a) `Yule10`

(b) `Yule20`

Fig. 5: Heatmap showing the majority wins based on MAE with simulations in five entropy categories (higher means noisier/harder); brighter colors mean a larger wining margin.

We also observe the regimes when we look at the mean relative error (MRE) of tree probabilities; see Fig. 6. (Since the results look very similar for `Yule10` and `Yule20`, those for `Yule10` can be found in Appendix B.) Note that for the MRE, a small absolute difference in probability for low probability trees causes a larger relative error. Since tree probabilities in the tail of the distribution are not that well estimated, we consider thus only the trees in the 50% and 90% credibility intervals. For small sample sizes, CCD0 performs better/equal than CCD1 up to about subsample sizes of 30/300. Note that CCD0 then does not improve any further, indicating the limitations of the CCD0 model. The performance of CCD1 remains the best even for larger subsample sizes with the sample distribution only slowly catching up.

Looking at the mean estimated rank of the top tree of the golden distribution in the other distributions for each simulation reveals a similar picture; see Fig. 7. CCD0 is best for subsample sizes up to and including 30, but above 100 CCD1 performs better on average; the sample distribution requires 1k samples to become competitive.

Precision. To evaluate the precision, we computed the difference in the tree probabilities between the two runs of each subsimulation. The mean over the 100 simulations for `Yule20` are shown in Fig. 8. We observe that CCD0 and CCD1 consistently show a higher precision than the sample distribution for all sample sizes. Note that high precision also implies a high stability.

Representativeness. Note that by construction for a given MCMC run, if the sample distribution contains the true tree then so do CCD1 and CCD0; analo-

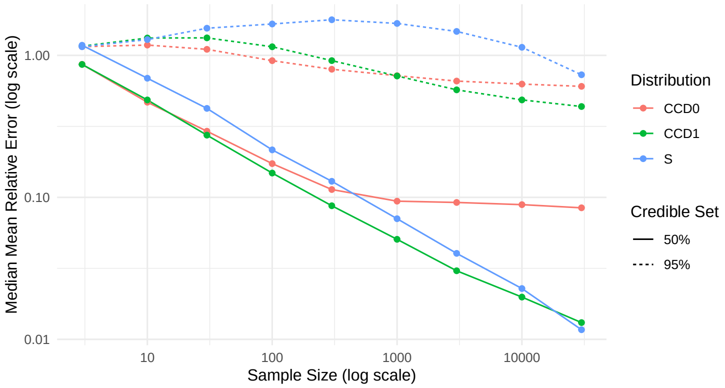


Fig. 6: Median MRE on the trees in the 50% and 90% percentile per sample size for `Yule20`.

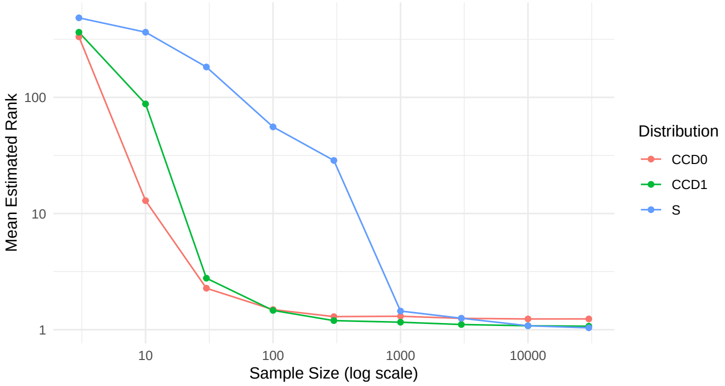


Fig. 7: Mean rank of the golden mode tree in the other distributions per sample size for `Yule20`.

gously, if CCD1 contains the true tree then so does CCD0. Table 1 shows how the percentage of the distributions (both runs per simulation) that contain the true tree for the 250/100 simulations of the `Yule20` and `Yule50` dataset. For the former, we observe that CCD0 and CCD1 cross the 95% threshold already for 100 samples, while the sample distribution only does so at 3k samples. The difference becomes even more apparent for `Yule50`, where the sample distribution only reaches 3.5% with 30k sampled trees, while CCD0 and CCD1 quickly contain the true tree in the majority of simulations and also reach the 95% threshold.

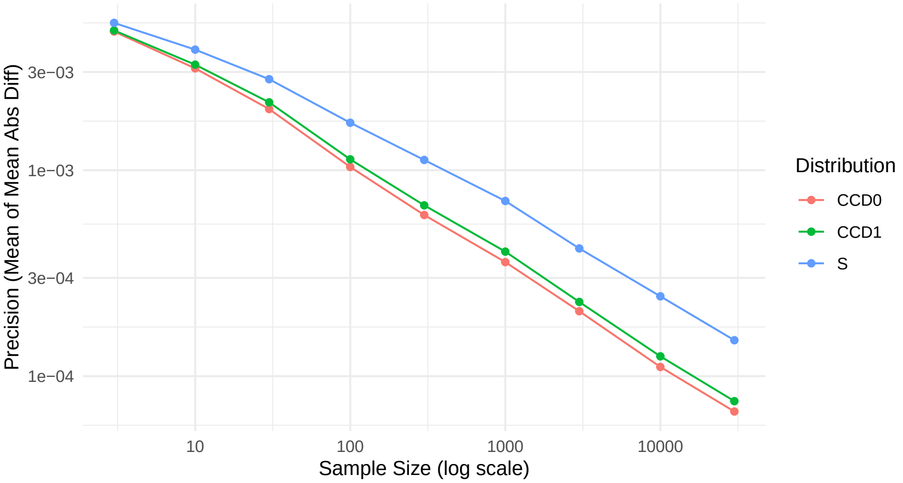


Fig. 8: Evaluating the precision of the distributions, we computed the mean of mean absolute differences of tree probabilities by the distributions between two runs per sample size for `Yule20`.

Table 1: Percentage of the true tree being contained in a distribution for `Yule20` and `Yule50` (out of the 250/100 simulations with 2 runs each).

		Sample Size								
	Distributions	3	10	30	100	300	1k	3k	10k	30k
`Yule20`	S	23.2	41	60.6	76.6	87.6	92.8	95.6	98.2	98.8
	CCD1	37	67.8	86.8	95.6	98.4	99.6	100	100	100
	CCD0	39.8	75.8	90.8	96.8	99.2	99.8	100	100	100
`Yule50`	S	0	0	0	0	0	0	1.5	2.5	3.5
	CCD1	0	3.5	24.5	50.5	69.5	80.5	90	98	99.5
	CCD0	0	9.5	46	70.5	87.5	94.5	100	99.5	100

3.2 Point Estimators

We evaluated the point estimators based on the following properties. Firstly, we have the *accuracy* – a good point estimate should be close to the truth (low Root Mean Squared Error or average distance; testable in simulation studies). Further, we can measure the behaviour under different MCMC runs; a good point estimator should be *precise* (small distance between estimates) and, related to that, *stable* (consistent distance to truth).

Holder et al. [15] argued for MRC trees as point estimates by showing that if we define a loss function with penalties for missed and wrong clades, then the MRC tree tries to minimize the loss. In fact, if we only report fully-resolved trees, then this is equivalent to the well-known Robinson-Foulds (RF) distance [21]. Recall that the *Robinson-Foulds (RF) distance* of two trees T and T' equals the symmetric distance of their clade sets $\mathcal{C}(T)$ and $\mathcal{C}(T')$. Basically, the RF distance measures how many clades the point estimate get wrong.

For our experiments, we used the datasets `Yule50` to `Yule400` and `Coal40` to `Coal320`. For each simulation and each of the two runs, we again used subsamples of size 3, 10, 30, 100, 300, 1k, 3k, 10k, and 30k to generate a CCD0, a CCD1, and a sample distribution each. With the CCDs we computed the CCD-MAP trees and the CCD0-MSCC tree, and based on the sample distribution we computed the MCC tree and the greedy consensus tree. The CCD0-MSCC tree behaved almost exactly as the CCD0-MAP tree and we thus excluded it from the figures to improve visual clarity. As reference we have the *true tree*, the one used to generate the alignments, of each simulation. (We only show the results for the for larger datasets here; those for the four smaller datasets are very similar and thus only given in Appendix C.)

Accuracy. Figure 9 shows the mean relative RF distance of the point estimates to the true tree for different subsample sizes. The relative RF distance describes the percentage of the $n-2$ clades of the true tree an estimator got wrong. So for e.g. `Yule400`, a relative RF distance of 0.08 (0.1) means that about 32 (resp. 40) of 398 nontrivial clades are different from the true tree. We observe that CCD0-MAP performs best from 3 to 30k trees. At around 30 to 100 trees for the Yule simulations and around 100 to 300 trees for the coalescent simulations greedy consensus catches up and performs equally well. CCD1-MAP gets close to this performance but does not fully catch up. MCC on the other remains at least 1% behind the top estimators.

Precision. To evaluate the precision, we computed the mean distance between the point estimates of two corresponding runs; see Fig. 10. We observe that greedy consensus and the CCD based methods have significant higher precision than MCC, with CCD1-MAP lacking slightly behind the others. For 1k trees, the CCD0 estimators and greedy consensus vary in less than 10 clades between runs, whereas MCC varies five- to tenfold of that. Note that a high precision also implies a high stability (variance in distance to the true tree).

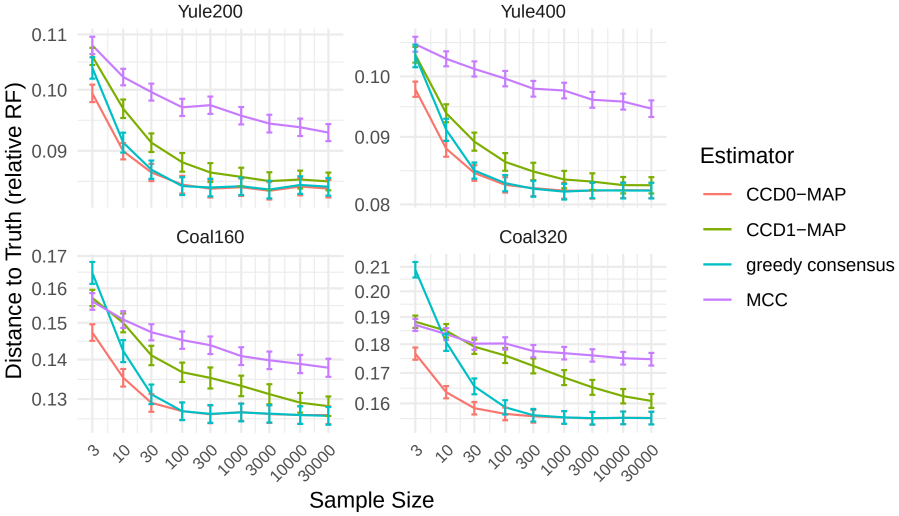


Fig. 9: The accuracy of the point estimates measured in terms of the mean relative RF distance to the true tree for different sample sizes of the large datasets.

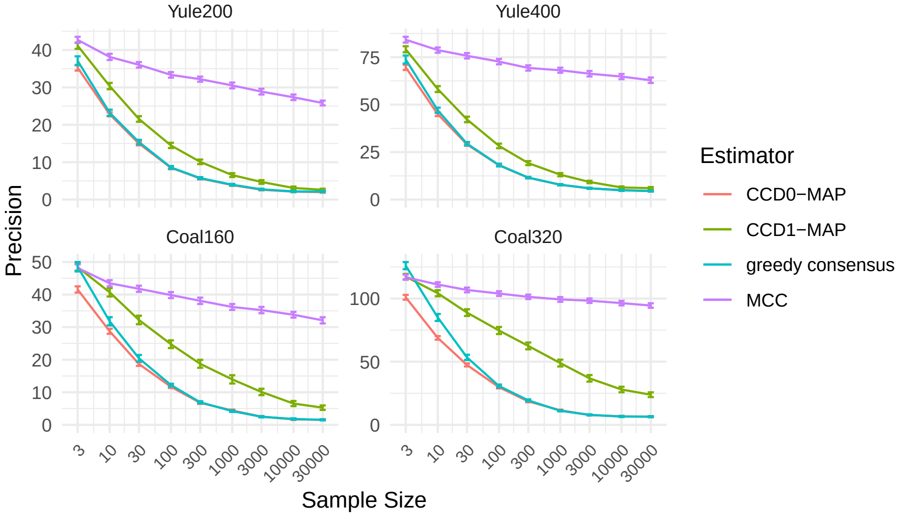


Fig. 10: The precision of the point estimates in terms of the RF distance, that is, the mean RF distance of the point estimates of the two runs of each simulation.

Running time. We also want to report on the running times of our implementations for the largest dataset Yule400. Constructing a CCD1 and CCD0 on subsamples of 30k trees (which requires parsing the file with 35k trees) took on average 90 seconds, the same as constructing the MCC tree. Computing any of the other point estimates took only few milliseconds. The bottleneck seems to be parsing the large file and not the construction.

Resolvedness. We also tested in how many simulations the greedy consensus tree was not fully resolved. The results in Table 2 show that for 300 trees and more, the greedy consensus tree was always fully resolved on our datasets. Note that with better Monte Carlo estimates of clade frequencies, ties that can cause unresolved trees become less likely.

Table 2: Number of unresolved greedy consensus trees out of 200 per dataset and sample size.

	Sample Size								
Dataset	3	10	30	100	300	1k	3k	10k	30k
Yule50	13	3	1	0	0	0	0	0	0
Yule100	24	8	1	0	0	0	0	0	0
Yule200	40	6	3	0	0	0	0	0	0
Yule400	59	21	5	1	0	0	0	0	0
Coal40	3	0	0	0	0	0	0	0	0
Coal80	23	3	0	0	0	0	0	0	0
Coal160	76	31	9	0	0	0	0	0	0
Coal320	162	92	9	2	0	0	0	0	0

4 Discussion

The CCD approach can be described as a “bias-variance trade-off” in the context of MCMC summarization. These tractable tree distributions exhibit a certain level of bias (due to the independence assumptions employed) in exchange for reduced variance in the estimates when faced with Monte Carlo error, particularly in cases of low ESS (relative to the posterior variance in tree space). The number of parameters of the models grow from CCD0 (clades) to CCD1 (clade splits) and finally to sample distributions (trees), demanding an increasing number of trees to estimate them accurately. This is confirmed by our experiments on easy and small problems, where we observed these three regimes in terms of the number of sampled trees: First, CCD0 is best for few samples in terms of accuracy, precision, and stability, then CCD1 catches up and becomes the best method in the mid range, while the sample distribution requires a huge number of sampled trees to become competitive. Unsurprisingly, the bias of CCD0 becomes apparent with a large enough number of uncorrelated samples. However, sampling enough trees with MCMC to reach the second and third regime is not feasible for non-trivial problems. Hence, CCD0 offers the overall best posterior estimate for most problems in practice.

With an implementation that uses CCD graphs, many tasks related to tree distributions can be performed efficiently (fixed-parameter tractable in the number of clades and clade splits). This includes sampling a tree, computing the probability of a tree, as required for example for BCA [1, 18], computing the MAP tree, and computing the entropy and the number of trees in the distribution. In practice, the running time is dominated by parsing the trees while building the CCD, whereas computing the MAP tree takes negligible time. While for large and very diffuse (more prior-like) distributions the construction of a CCD0 may take noticeable time (minutes), it would still only be a fraction of the days or weeks needed to compute such a distribution via MCMC.

Concerning the point estimates, we demonstrated that the CCD-MAP trees and the greedy consensus tree outperform the commonly used MCC tree in terms of accuracy and precision. So not only do they produce better trees in general, but they are also more robust to the random sampling process of MCMC. This finding is concerning given that the MCC tree has been the standard point estimate used by almost every BEAST practitioner for decades. Additionally, we find that the CCD0-MAP tree performs equally or better than the greedy consensus tree, with the added benefit that both variants of the CCD-MAP tree guarantee a fully resolved tree. While getting an unresolved greedy consensus tree may not be an issue for many problems (cf. Table 2), we want to point out that (ii) in viral phylodynamics, it is typical to encounter (near)identical sequences resulting in partially diffuse posteriors, thus increasing the probability of encountering unresolved greedy consensus trees, and (ii) finding the most resolved greedy consensus tree is an NP-hard problem. The CCD1-MAP tree does not match the accuracy of the CCD0-MAP tree in our experiments on nontrivial problems, since even for large samples we do not reach the CCD1-regime observed in smaller analyses. On the `Yule20` dataset, we could not observe a performance difference between the CCD1-MAP tree and CCD0-MAP tree. For a sufficiently large number of uncorrelated samples, the CCD1-MAP tree is expected to perform equal or even better than the CCD0-MAP tree.

Despite the existence of various tree metrics our evaluation focuses on the Robinson-Foulds distance. This choice is justified because all the point estimates compared in the paper – CCD-MAP, MCC, and greedy consensus – are primarily based on constructing a topology. Hence, the Robinson-Foulds distance is particularly suitable for evaluating their performance, especially in the context of systematics, where one of the primary goals of a phylogeny is to obtain accurate clade information [15]. In this context, the Robinson-Foulds metric directly quantifies the performance when comparing a point estimate to the true tree.

5 Conclusion

This research has shown that the CCD0-MAP tree and the greedy consensus should be the preferred point estimators for Bayesian phylogenetic inference of time-trees. The restriction to sampled trees comes at such a high cost that previous caution of using unsampled trees as point estimates is not warranted. Furthermore, CCDs offer better estimates of individual tree probabilities than the sample distribution. We can thus retire the MCC-from-sample point estimator.

While our approach was developed mainly for TreeAnnotator within the BEAST2 framework [6], our results are applicable to any sample of rooted tree topologies that represents a posterior distribution. We have incorporated CCD-based point estimators into the existing TreeAnnotator software enabling users to easily access and use this new method on their data.[4]

In practice, time information of point estimates is also of great interest. The CCD-based point estimates fit in the commonly used framework of estimating the tree topology first followed by annotating it with divergence ages. These latter methods are independent from CCDs. It would be interesting to see how greedy consensus and the CCD0-MAP tree combined with an annotation method perform in comparison to other combined approaches and to methods that estimate the topology and branch lengths at the same time, like the matrix method [7].

We hope to use and further develop CCDs for other tasks when working with posterior distributions. This includes the computation of the credibility set of tree topologies, MCMC convergence analysis (cf. Berling et al. [3]), and detection of rogue taxa.

Acknowledgements

LB, JK, AG, and AJD were partially supported by the Beyond Prediction Data Science Research Programme (MBIE grant UOAX1932). AG was partially supported by a Marsden Grant from the Royal Society Te Apārangi 21-UOC-057. We would also like to thank Jordan Douglas for his help setting up simulation studies.

References

1. Cécile Ané, Bret Larget, David A. Baum, Stacey D. Smith, and Antonis Rokas. Bayesian estimation of concordance among gene trees. *Molecular Biology and Evolution*, 24(2):412–426, 11 2006. doi:10.1093/molbev/msl170.
2. Philipp Benner, Miroslav Bačák, and Pierre-Yves Bourguignon. Point estimates in phylogenetic reconstructions. *Bioinformatics*, 30(17):i534–i540, 2014. doi:10.1093/bioinformatics/btu461.

[4] A new version of TreeAnnotator, which includes the CCD-MAP tree, and the code will be made available after the peer-review process.

3. Lars Berling, Remco Bouckaert, and Alex Gavryushkin. Automated convergence diagnostic for phylogenetic MCMC analyses. *bioRxiv*, 2023. doi:10.1101/2023.08.10.552869.
4. Lars Berling, Lena Collienne, and Alex Gavryushkin. Estimating the mean in the space of ranked phylogenetic trees. *bioRxiv*, 2023. doi:10.1101/2023.05.08.539790.
5. Louis J. Billera, Susan P. Holmes, and Karen Vogtmann. Geometry of the space of phylogenetic trees. *Advances in Applied Mathematics*, 27(4):733–767, 2001. doi:10.1006/aama.2001.0759.
6. Remco Bouckaert, Timothy G Vaughan, Joëlle Barido-Sottani, Sebastián Duchêne, Mathieu Fourment, Alexandra Gavryushkina, Joseph Heled, Graham Jones, Denise Kühnert, Nicola De Maio, et al. BEAST 2.5: An advanced software platform for Bayesian evolutionary analysis. *PLoS computational biology*, 15(4):e1006650, 2019. doi:10.1371/journal.pcbi.1006650.
7. Remco R Bouckaert. Variational bayesian phylogenies through matrix representation of tree space. *bioRxiv*, pages 2023–10, 2023.
8. Daniel G Brown and Megan Owen. Mean and Variance of Phylogenetic Trees. *Systematic Biology*, 69(1):139–154, 2019. doi:10.1093/sysbio/syz041.
9. David Bryant. A classification of consensus methods for phylogenetics. *DIMACS series in discrete mathematics and theoretical computer science*, 61:163–184, 2003.
10. Alexei J Drummond, Kylie Chen, Fábio K Mendes, and Dong Xie. LinguaPhylo: a probabilistic model specification language for reproducible phylogenetic analyses. *PLOS Computational Biology*, 19(7):e1011226, 2023. doi:10.1371/journal.pcbi.1011226.
11. Alex Gavryushkin and Alexei J Drummond. The space of ultrametric phylogenetic trees. *Journal of theoretical biology*, 403:197–208, 2016. doi:10.1016/j.jtbi.2016.05.001.
12. Pawel Gawrychowski, Gad M. Landau, Wing-Kin Sung, and Oren Weimann. A faster construction of greedy consensus trees. In Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Donald Sannella, editors, *International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 63:1–63:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.63.
13. Masami Hasegawa, Hirohisa Kishino, and Taka-aki Yano. Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *Journal of molecular evolution*, 22:160–174, 1985. doi:10.1007/BF02101694.
14. Joseph Heled and Remco R Bouckaert. Looking for trees in the forest: summary tree from posterior samples. *BMC Evolutionary Biology*, 13:221, 2013. doi:10.1186/1471-2148-13-221.
15. Mark T. Holder, Jeet Sukumaran, and Paul O. Lewis. A justification for reporting the majority-rule consensus tree in Bayesian phylogenetics. *Systematic Biology*, 57(5):814–821, 2008. doi:10.1080/10635150802422308.
16. Sebastian Höhna and Alexei J. Drummond. Guided tree topology proposals for Bayesian phylogenetic inference. *Systematic Biology*, 61(1):1–11, 2012. doi:10.1093/sysbio/syr074.
17. Sebastian Höhna, Michael J. Landis, Tracy A. Heath, Bastien Boussau, Nicolas Lartillot, Brian R. Moore, John P. Huelsenbeck, and Fredrik Ronquist. RevBayes: Bayesian phylogenetic inference using graphical models and an interactive model-specification language. *Systematic Biology*, 65(4):726–736, 2016. doi:10.1093/sysbio/syw021.

18. Bret Larget. The estimation of tree posterior probabilities using conditional clade probability distributions. *Systematic Biology*, 62(4):501–511, 2013. doi:10.1093/sysbio/syt014.

19. Paul O. Lewis, Ming-Hui Chen, Lynn Kuo, Louise A. Lewis, Karolina Fučíková, Suman Neupane, Yu-Bo Wang, and Daoyuan Shi. Estimating Bayesian phylogenetic information content. *Systematic Biology*, 65(6):1009–1023, 2016. doi:10.1093/sysbio/syw042.

20. Fabio K Mendes, Remco Bouckaert, Luiz M Carvalho, and Alexei J Drummond. How to validate a bayesian evolutionary model. *bioRxiv*, pages 2024–02, 2024.

21. D. F. Robinson and L. R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(1):131–147, 1981. doi:10.1016/0025-5564(81)90043-2.

22. Allen G Rodrigo and Joseph Felsenstein. Coalescent approaches to HIV population genetics. *The evolution of HIV*, pages 233–272, 1999.

23. Fredrik Ronquist, Maxim Teslenko, Paul van der Mark, Daniel L. Ayres, Aaron Darling, Sebastian Höhna, Bret Larget, Liang Liu, Marc A. Suchard, and John P. Huelsenbeck. MrBayes 3.2: efficient Bayesian phylogenetic inference and model choice across a large model space. *Systematic Biology*, 61(3):539–542, 2012. doi:10.1093/sysbio/sys029.

24. Wing-Kin Sung. Greedy consensus tree and maximum greedy consensus tree problems. In Gautam K. Das, Partha S. Mandal, Krishnendu Mukhopadhyaya, and Shin-ichi Nakano, editors, *WALCOM: Algorithms and Computation*, volume 11355 of *LNCS*, pages 305–316. Springer, 2019. doi:0.1007/978-3-030-10564-8_24.

25. Gergely J. Szöllősi, Wojciech Rosikiewicz, Bastien Boussau, Eric Tannier, and Vincent Daubin. Efficient exploration of the space of reconciled gene trees. *Systematic Biology*, 62(6):901–912, 2013. doi:10.1093/sysbio/syt054.

26. Hongxun Wu. Near-optimal algorithm for constructing greedy consensus tree. In Artur Czumaj, Anuj Dawar, and Emanuela Merelli, editors, *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*, volume 168 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 105:1–105:14. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.ICALP.2020.105.

27. George Udny Yule. II.—A mathematical theory of evolution, based on the conclusions of Dr. JC Willis, FR S. *Philosophical Transactions of the Royal Society of London, Series B*, 213(402-410):21–87, 1925. doi:10.1098/rstb.1925.0002.

Appendix

A Graphical Models of Datasets

```
data {
   L = 300;
   clockRate = 1.0;
   nCatGamma = 4;
   birthRate = 25.0;
   n = 50;
}
model {
   frequencies ∼ Dirichlet(conc=[5.0, 5.0, 5.0, 5.0]);
   kappa ∼ LogNormal(meanlog=1.0, sdlog=1.25);
   Q = hky(kappa=kappa, freq=frequencies);
   shape ∼ LogNormal(meanlog=-1.0, sdlog=0.5);
   siteRates ∼ DiscretizeGamma(shape=shape, ncat=nCatGamma, replicates=L);
   phi ∼ Yule(lambda=birthRate, n=n);
   D ∼ PhyloCTMC(L=L, Q=Q, mu=clockRate, siteRates=siteRates, tree=phi);
}
```

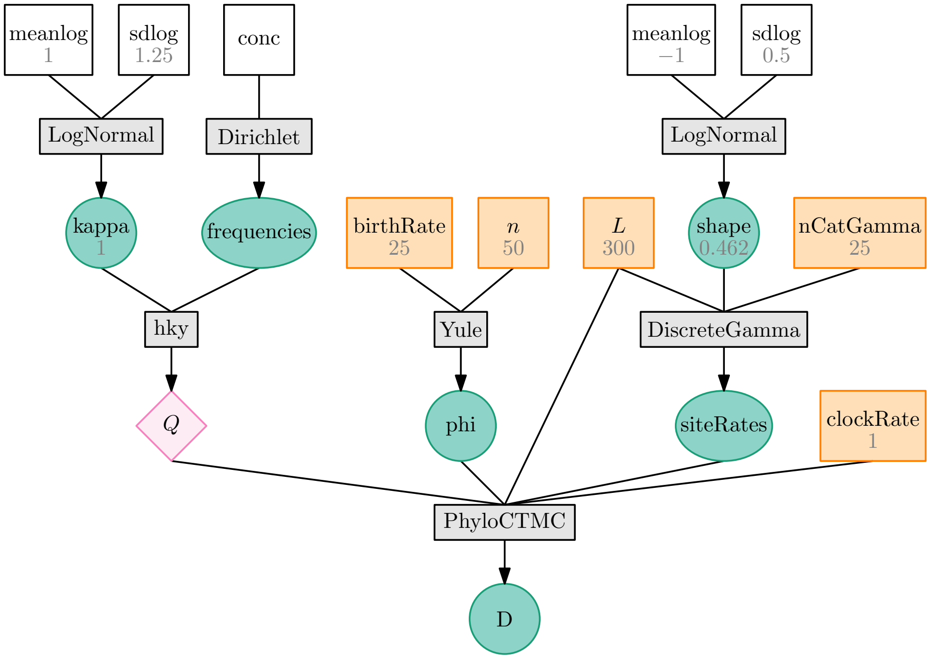


Fig. 11: lphy script and graphical model of Yule datasets.

```
data {
   L = 250;
   clockRate = 1.0;
   nCatGamma = 4;
   n = 40;
}
model {
   frequencies ~ Dirichlet(conc=[5.0, 5.0, 5.0, 5.0]);
   kappa ~ LogNormal(meanlog=1.0, sdlog=1.25);
   Q = hky(kappa=kappa, freq=frequencies);
   shape ~ LogNormal(meanlog=-1.0, sdlog=0.5);
   siteRates ~ DiscretizeGamma(shape=shape, ncat=nCatGamma, replicates=L);
   positiveAges ~ Uniform(lower=0, upper=0.2, replicates=n-1);
   leafAges = concatArray([0.0], positiveAges);
   popSize ~ LogNormal(meanlog=-2.4276, sdlog=0.5);
   phi ~ Coalescent(ages=leafAges, n=n, theta=popSize);
   D ~ PhyloCTMC(L=L, Q=Q, mu=clockRate, siteRates=siteRates, tree=phi);
}
```

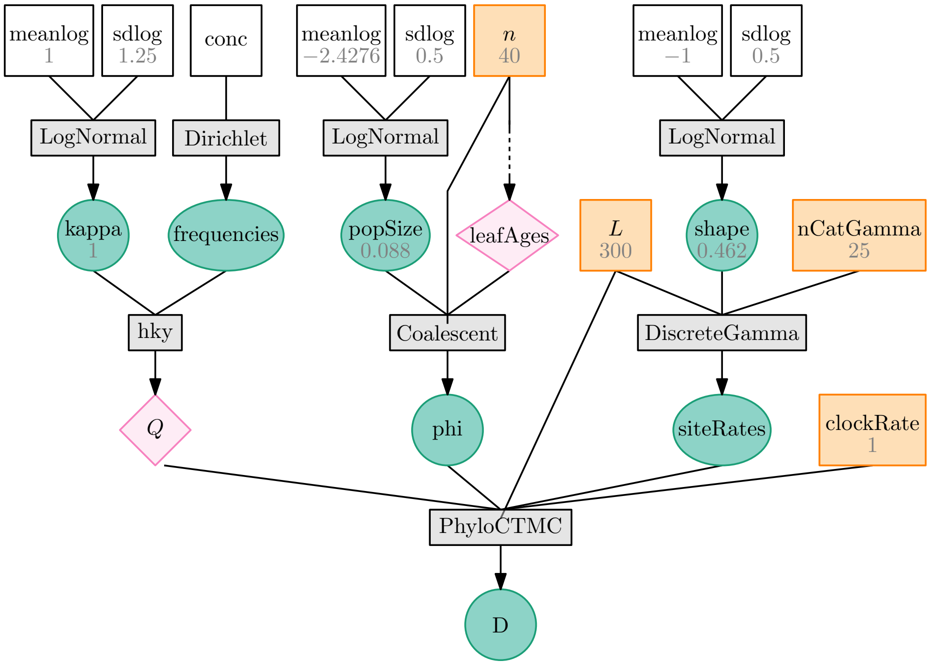


Fig. 12: lphy script and graphical model of Coalescent datasets.

B Further Results on the Distributions

The mean MRE of tree probabilities of the trees in the 75% and 90% credibility 640 intervals for Yule10 are shown in Figure 13 showing the same results as for Yule20 in Fig. 6.

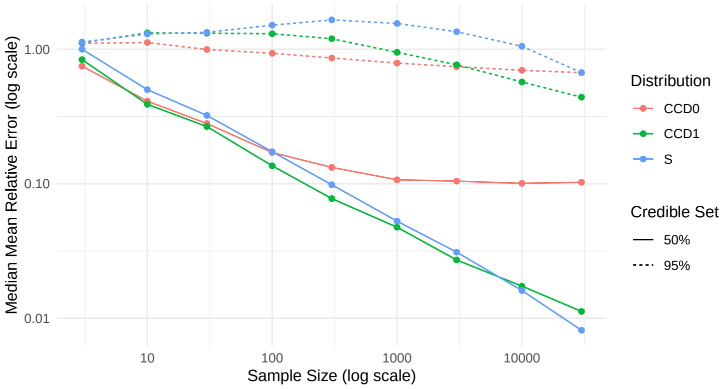


Fig. 13: Mean Relative Error (MRE) on the trees in the 50% and 90% percentile per sample size for Yule10.

Figure 14 shows the mean estimated rank of the top tree of the golden distribution in the other distributions for Yule10. One difference to the Yule20 results in Fig. 7 is that the difference between the sample distribution (CCD1) 645 and CCD0 is smaller (larger) not perform as badly for sample sizes up to 300 trees, but overall the same tendencies emerge.

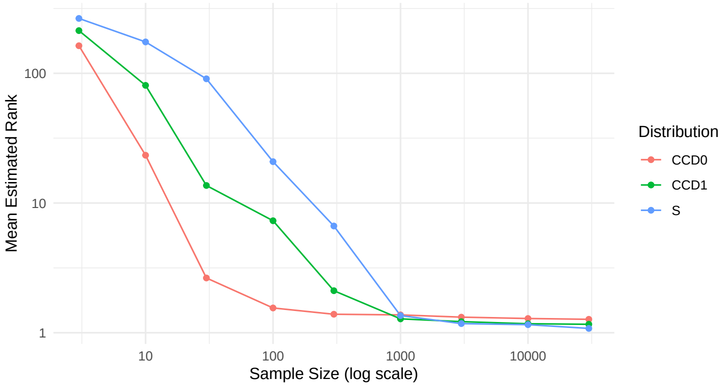


Fig. 14: Mean rank of the golden mode tree in the other distributions per sample size for Yule10.

C Further Results on Point Estimators

For completeness, we provide here the accuracy and precision results of the point estimators for the smaller datasets Yule50, Yule100, Coal40, and Coal80. The
650 accuracy results are shown in Fig. 15 (cf. Fig. 9) and the precision results are shown in Fig. 16 (cf. Fig. 10).

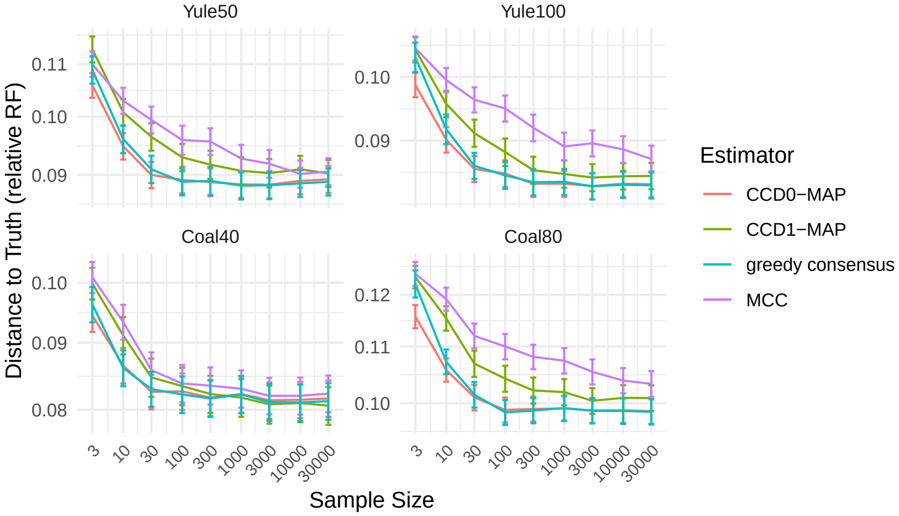


Fig. 15: The accuracy of the point estimates measured in terms of the mean relative RF distance to the true tree for different sample sizes of the large datasets.

In addition, we also computed the stability of the point estimates, that is, the mean difference between the distances to the true tree between two corresponding runs; see Fig. 17. We observe that the CCD-MAPs and greedy consensus are more
655 stable than MCC, which is not surprising given their higher precision.

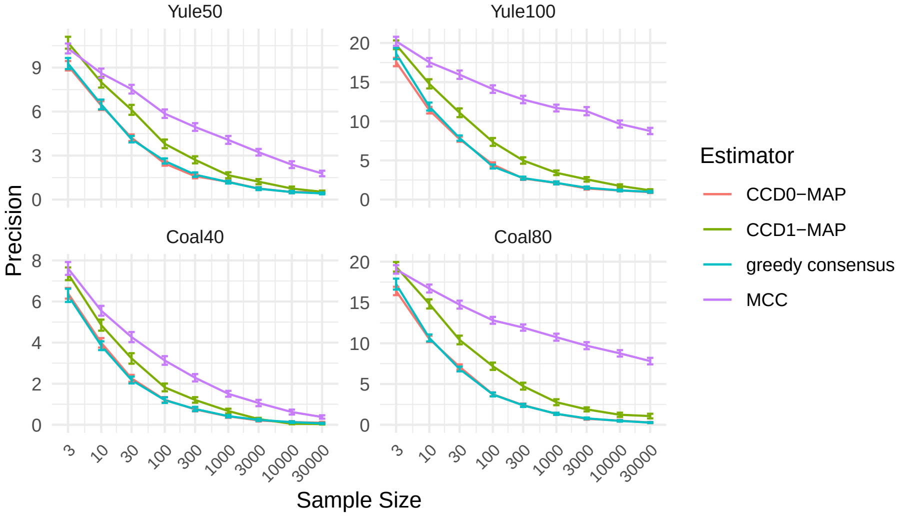


Fig. 16: The precision of the point estimates in terms of the RF distance, that is, the mean RF distance of the point estimates of the two runs of each simulation.

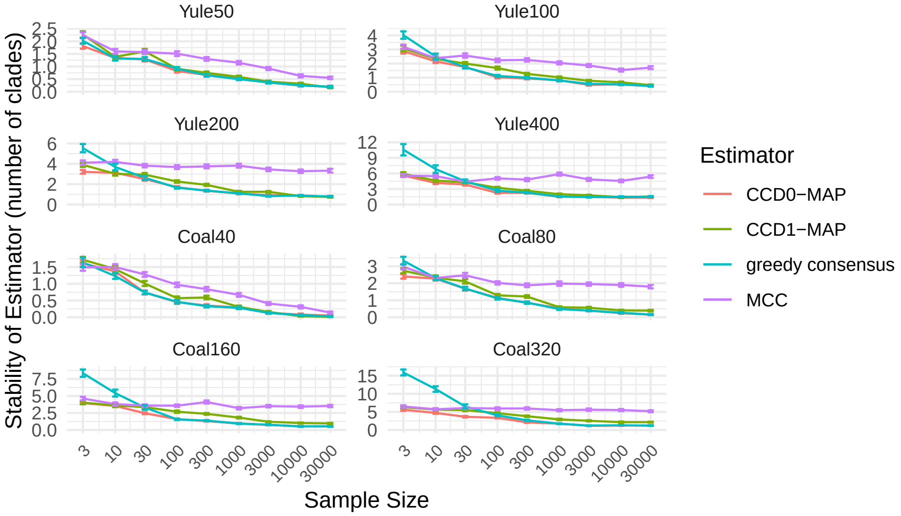


Fig. 17: The stability of the point estimates in terms of the RF distance, that is, the mean difference of RF distance of the point estimate to the truth of the two runs of each simulation.